UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/503,215	02/12/2000	Jack Niewiadomski	777.364US1	6252

7590 12/22/2004

John E Whitaker
Merchant & Gould PC
P O Box 2903
Minneapolis, MN 55402-0903

EXAMINER	
NAHAR, QAMRUN	
ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 12/22/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/503,215	NIEWIADOMSKI ET AL.	
	Examiner	Art Unit	
	Qamrun Nahar	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 November 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-26,28-33,36 and 41-47* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-26, 28-33, 36 and 41-47* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the RCE filed on 11/15/04.

2. Claims 1, 14, 15, 21, 23, 24, 28-32, 36, 41 and 46 have been amended.

3. Claims 1-26, 28-33, 36 and 41-47 are pending.

4. Claims 1-20, 28-33, 36 and 41-47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fortin (U.S. 5,528,753) in view of Kaufer (U.S. 5,812,828).

5. Claims 21-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fortin (U.S. 5,528,753) in view of Kaufer (U.S. 5,812,828), and further in view of Peek (U.S. 5,481,706).

Response to Amendment

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-20, 28-33, 36 and 41-47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fortin (U.S. 5,528,753) in view of Kaufer (U.S. 5,812,828).

Per Claim 1 (Amended):

Fortin teaches a computerized method for creating an instrumented executable file ("provide a system and method for efficiently instrumenting stripped object routines executing on a computer system" in column 3, lines 19-21); modifying an executable file to invoke a user-

supplied function in place of an original function, the user-supplied function enabling code to control execution of the original function ("The Call to Target still points to address 202. After 202, however, control is passed to Entry Routine 210. Entry Routine 210 collects the information desired by the monitor and returns control to the target routine." in column 4, lines 46-51); retaining access information of the original function in a function lookup table; retrieving the access information from the function lookup table using the name of the original function; and invoking the original function using the access information ("The Call to Target still points to address 202 ... returns control to the target routine." in column 4, lines 47-51; "demultiplexor entries" in column 5, lines 46-67; column 6, lines 6-9; and column 6, lines 35-39 and lines 53-58). Fortin does not explicitly teach that the code is fault simulation code. Kaufer teaches fault simulation code (column 1, lines 59-64).

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by Fortin to include fault simulation code using the teaching of Kaufer. The modification would be obvious because one of ordinary skill in the art would be motivated to simulate one or more errors during testing of an application (Kaufer, column 1, lines 41-44).

Per Claim 2:

The rejection of claim 1 is incorporated, and Fortin further teaches the user-supplied function is modified to invoke the original function using the retained access information of the original function (column 4, lines 49-53).

Per Claim 3:

The rejection of claim 1 is incorporated, and the combination of Fortin and Kaufer further teaches that the user-supplied function is in a library (Fortin, column 5, lines 20-24). The combination of Fortin and Kaufer does not explicitly teach that the user-supplied function is in a dynamic link library. Official Notice is taken that it was a common practice to have functions in a dynamic link library.

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by the combination of Fortin and Kaufer to include that the user-supplied function is in a dynamic link library using the teaching of common practice. The modification would be obvious because one of ordinary skill in the art would be motivated to modify functions without recompiling the program.

Per Claim 4:

The rejection of claim 1 is incorporated, and Fortin further teaches the user-supplied function is not exported during compilation (column 5, lines 20-24).

Per Claim 5:

The rejection of claim 1 is incorporated, and Fortin further teaches the original function and the user-supplied function have identical prototypes (column 4, lines 62-65).

Per Claim 6:

The rejection of claim 1 is incorporated, and Fortin further teaches the user-supplied function is stored in a module that is separate from the executable file (column 5, lines 20-24).

Per Claim 7:

The rejection of claim 1 is incorporated, and Fortin further teaches modifying the executable file is performed using user-specified set points (column 4, lines 54-65).

Per Claim 8:

The rejection of claim 7 is incorporated, and Fortin further teaches determining whether the original function implements the thiscall calling convention, and when the determination is positive, adding instructions to the executable file to perform pushing the register that holds the "this" pointer onto the stack from the invoked original function site when the determining indicates that the function implements a thiscall calling convention (column 5, lines 48-51 and lines 55-57, and column 6, lines 48-58); and swapping the return value of the invoking original function on the stack and the register that holds the "this" pointer value on the stack when the determining indicates that the function implements a thiscall calling convention (column 6, lines 59-67).

Per Claim 9:

The rejection of claim 7 is incorporated, and Fortin further teaches modifying the executable file further comprises enabling the user-supplied function to invoke the original function in the executable file (column 4, lines 49-53).

Per Claim 10:

The rejection of claim 9 is incorporated, and Fortin further teaches adding a jump in the user-supplied function to a function that retrieves the address of the original function (column 6, lines 6-18); and adding a jump in the user-supplied function that invokes the original function using the address of the original function (column 6, lines 1-5 and lines 35-39).

Per Claim 11:

The rejection of claim 1 is incorporated, and Fortin further teaches enabling the user-supplied function to alter behavior (column 4, lines 47-53).

Per Claim 12:

The rejection of claim 11 is incorporated, and Fortin further teaches enabling the user-supplied function to alter behavior is performed in response to data (column 4, lines 47-53).

Per Claim 13:

The rejection of claim 12 is incorporated, and Fortin further teaches the data is retrieved from an initialization file ("libinstr.a" in Fig. 7 and column 5, lines 20-24).

Per Claim 14 (Amended):

The rejection of claim 1 is incorporated, and Fortin further teaches saving the address of the original function in a threaded local storage variable (column 6, lines 1-5); and creating an

entry in the function lookup table associating the address of the original function with the name of the original function, wherein the function lookup table is in the instrumented executable file (column 5, lines 46-51 and lines 64-67).

Per Claim 15 (Amended):

Fortin teaches a computerized method for executing an instrumented executable file ("provide a system and method for efficiently instrumenting stripped object routines executing on a computer system" in column 3, lines 19-21); modifying the instrumented executable file to invoke a user-supplied function in place of an original function, the user-supplied function enabling code to control execution of the original function, the user-supplied function having a jump to the original function ("The Call to Target still points to address 202. After 202, however, control is passed to Entry Routine 210. Entry Routine 210 collects the information desired by the monitor and returns control to the target routine. ... The stored target routine first instruction is executed in the 'Return to Target' section and control branches to the second instruction of the target routine 714." in column 4, lines 46-51 and column 6, lines 56-58); saving the address of the original function in a threaded local storage variable ("The 'Return to Target Routine' section 604 contains the 'saved' first instruction of the target routine and direct branch to the address of the second instruction of the target routine." in column 6, lines 1-5); retrieving the address from the threaded local storage variable using the name of the original function; and invoking the user-supplied function using the address ("Processing begins with a call to the target routine 702. This call will encounter the branch first instruction and immediately branch to the Entry section 704 ... that in turn calls 708 user supplied Entry

routine" in column 6, lines 46-51; column 6, lines 6-9; and column 6, lines 35-39 and lines 53-58). Fortin does not explicitly teach that the code is fault simulation code. Kaufer teaches fault simulation code (column 1, lines 59-64).

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by Fortin to include fault simulation code using the teaching of Kaufer. The modification would be obvious because one of ordinary skill in the art would be motivated to simulate one or more errors during testing of an application (Kaufer, column 1, lines 41-44).

Per Claim 16:

The rejection of claim 15 is incorporated, and Fortin further teaches creating a master lookup table at initialization wherein the master lookup table associates the base address of the instrumented executable file to the address of a function lookup table in the instrumented executable file (column 5, lines 47-51 and lines 55-67).

Per Claim 17:

The rejection of claim 15 is incorporated, and the combination of Fortin and Kaufer further teaches that the saving and the invoking are performed by the original function, and the original function being located in the instrumented executable file (Fortin, column 6, lines 44-51). The combination of Fortin and Kaufer does not explicitly teach that the original function is in a dynamic link library and a stub function of the original function being located in the

instrumented executable file. Official Notice is taken that it was a common practice to include a function in a dynamic link library at the time the instant invention was made.

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by the combination of Fortin and Kaufer to include a function in a dynamic link library using the teaching of common practice. The modification would be obvious because one of ordinary skill in the art would be motivated to eliminate recompilation of a file, save memory space, and one would also want to provide the ease of modifying functions.

Per Claim 18:

The rejection of claim 15 is incorporated, and Fortin further teaches wherein original function is embedded in the instrumented executable file, and wherein the saving and the invoking is performed by the original function (column 6, lines 46-51).

Per Claim 19:

The rejection of claim 15 is incorporated, and Fortin further teaches invoking the original function from within the user-supplied function using the threaded local storage variable (column 6, lines 56-58).

Per Claim 20:

The rejection of claim 19 is incorporated, and Fortin further teaches pushing the register that holds the "this" pointer onto the stack from the invoked original function site when the

determining indicates that the function implements a thiscall calling convention (column 5, lines 48-51 and lines 55-57, and column 6, lines 48-58); and swapping the return value of the invoking original function on the stack and the register that holds the "this" pointer value on the stack when the determining indicates that the function implements a thiscall calling convention (column 6, lines 59-67).

Per Claim 28 (Amended):

This is a system version of the claimed method discussed above, claim 1, wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also obvious.

Per Claim 29 (Amended):

This is a system version of the claimed method discussed above (claims 1 and 2), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above, including "the original function having an identity comprising a name and a parameter prototype" (Fortin, column 4, lines 62-65 and column 5, lines 64-67). Thus, accordingly, this claim is also obvious.

Per Claim 30 (Amended):

Fortin teaches a computerized system ("provide a system and method for efficiently instrumenting stripped object routines executing on a computer system" in column 3, lines 19-21); an executable file having a jump to an original function, the original function having an

identity comprising a name and a parameter prototype ("The stored target routine first instruction is executed in the "Return to Target" section and control branches to the second instruction of the target routine 714." in column 5, lines 64-67 to column 6, lines 1-5 and lines 56-58); a first software component having a user-supplied function that includes a jump to the original function ("The Entry routine saves ... and calls 706 the common Entry code that in turn calls 708 user supplied Entry routine ... The stored target routine first instruction is executed in the "Return to Target" section and control branches to the second instruction of the target routine 714." in column 5, lines 64-67 to column 6, lines 1-5 and lines 48-58); and a second software component for receiving the identity of the original function, receiving the identity of the user-supplied function, instrumenting the executable file by modifying the executable file to invoke the identity of the user-supplied function in place of the identity of the original function, the identity of the user-supplied function enabling code to control execution of the original function ("The routines also allow the function of a target routine to be fully replaced such that newly provided code will be executed instead of the base code in the routine being monitored." in column 4, lines 62-65); storing the original function address in the executable file in association with the name of the original instrumented function; retrieving the original function address using the name of the original instrumented function; and invoking the original function using the original function address ("insertion of the demultiplexor entries 510. A demultiplexor entry (demux-entry) is provided for each target routine ... The data section 602 contains the addresses at which the target routine and the target routine's symbolic name reside" in column 5, lines 46-51 and lines 64-67; column 6, lines 6-9; and column 6, lines 35-39 and lines 53-58). Fortin does not

explicitly teach that the code is fault simulation code. Kaufer teaches fault simulation code (column 1, lines 59-64).

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the system disclosed by Fortin to include fault simulation code using the teaching of Kaufer. The modification would be obvious because one of ordinary skill in the art would be motivated to simulate one or more errors during testing of an application (Kaufer, column 1, lines 41-44).

Per Claim 31 (Amended):

Fortin teaches a computerized system ("provide a system and method for efficiently instrumenting stripped object routines executing on a computer system" in column 3, lines 19-21); a first module of machine-readable code comprising a call to an original function, the call being directed to a user-supplied function ("The Call to Target still points to address 202. After 202, however, control is passed to Entry Routine 210. Entry Routine 210 collects the information desired by the monitor and returns control to the target routine." in column 4, lines 47-51); and a first data structure associating the identity of the original function with the location of the original function ("A demultiplexor entry (demux-entry) is provided for each target routine ... The data section 602 contains the addresses at which the target routine and the target routine's symbolic name reside" in column 5, lines 46-51 and lines 64-67); and a second module comprising the user-supplied function, linked to the first module and a jump to the original function, the user-supplied function enabling code to control execution of the original function, wherein the location of the original function is retrieved using the identity of the original

function, and wherein the original function is invoked using the location of the original function ("The stored target routine first instruction is executed in the 'Return to Target' section and control branches to the second instruction of the target routine 714." in column 6, lines 56-58; column 6, lines 6-9; and column 6, lines 35-39 and lines 53-58). Fortin does not explicitly teach that the code is fault simulation code. Kaufer teaches fault simulation code (column 1, lines 59-64).

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the system disclosed by Fortin to include fault simulation code using the teaching of Kaufer. The modification would be obvious because one of ordinary skill in the art would be motivated to simulate one or more errors during testing of an application (Kaufer, column 1, lines 41-44).

Per Claim 32 (Amended):

The rejection of claim 31 is incorporated, and Fortin further teaches the first data structure comprises a function lookup table for verifying that a threaded local storage variable contains the correct address for the original function (column 5, lines 64-67 to column 6, lines 1-5); and wherein the second module comprises a dynamic linked library (column 5, lines 20-24).

Per Claim 33:

The rejection of claim 31 is incorporated, and Fortin further teaches a second data structure associating the location of the first data structure with the location of the first module (column 5, lines 47-51 and lines 55-67).

Per Claim 36 (Amended):

This is a computer-readable medium version of the claimed method discussed above, claim 1, wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also obvious.

Per Claims 41 (Amended) & 43:

These are another versions of the claimed method discussed above, claim 1, wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above, including "replacing the access to the original function with an access to a user-supplied function" (Fortin, column 4, lines 46-51). Thus, accordingly, these claims are also obvious.

Per Claim 42:

This is another version of the claimed method discussed above, claim 2, wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also obvious.

Per Claim 44:

The rejection of claim 41 is incorporated, and Fortin further teaches wherein replacing the access to the original function with the access to the user-supplied function is performed by modifying set points stored in a computer-readable medium separate from the executable file (column 4, lines 54-65).

Per Claim 45:

The rejection of claim 41 is incorporated, and Fortin further teaches wherein retaining access information associated with the original function includes saving the address of the original function (column 6, lines 1-5).

Per Claim 46 (Amended):

The rejection of claim 41 is incorporated, and Fortin further teaches wherein retaining access information associated with the original function includes associating the name of the original function with the address of original function using the function lookup table (column 5, lines 46-51 and lines 64-67).

Per Claim 47:

The rejection of claim 46 is incorporated, and Fortin further teaches further comprising invoking the original function using the function lookup table (column 5, lines 46-51 and lines 64-67).

8. Claims 21-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fortin (U.S. 5,528,753) in view of Kaufer (U.S. 5,812,828), and further in view of Peek (U.S. 5,481,706).

Per Claims 21 (Amended) & 22:

Fortin teaches a computerized method for instrumenting an original function in an executable file for testing by callers of the original function ("provide a system and method for efficiently instrumenting stripped object routines executing on a computer system" in column 3, lines 19-21), the original function comprises an instruction that saves the addresss of the original function to a threaded local storage variable ("The 'Return to Target Routine' section 604 contains the 'saved' first instruction of the target routine and direct branch to the address of the second instruction of the target routine." in column 6, lines 1-5), replaces an access to the original function with an access to a user-supplied function, the user-supplied function enabling code to control execution of the original function ("Processing begins with a call to the target routine 702. This call will encounter the branch first instruction and immediately branch to the Entry section 704 ... that in turn calls 708 user supplied Entry routine" in column 6, lines 46-51), adding an entry in a function lookup table of the original function; retrieving the address of the original function from the threaded local storage variable using the name of the original function; and invoking the original function using the address ("insertion of the demultiplexor entries 510. A demultiplexor entry (demux-entry) is provided for each target routine." in column 5, lines 46-51 and lines 64-67; column 6, lines 6-9; and column 6, lines 35-39 and lines 53-58), determining if the prototype of the original function is correctly specified and indicating an error when the determining indicates an incorrectly specified prototype of the original function ("The routines also allow the function of a target routine to be fully replaced such that newly provided code will be executed instead of the base code in the routine being monitored." in column 4, lines 62-65, inherently determining prototype). Fortin does not explicitly teach an imported function, a stub

function for the imported function, adding a wrapper of the imported function to an import data block or that the code is fault simulation code.

Official Notice is taken that it was a common practice to include a stub function for an imported function at the time the instant invention was made. Kaufer teaches fault simulation code (column 1, lines 59-64). Peek teaches adding a wrapper of a function to a data block ("providing a "wrapper" for the particular function ... a wrapper will be seen to be an individual piece of code placed around a corresponding individual function call or routine" in column 6, lines 34-37 and 43-50).

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by Fortin to include a stub function for an imported function, adding a wrapper of the imported function to an import data block, and fault simulation code using the combined teaching of common practice, Kaufer and Peek. The modification would be obvious because one of ordinary skill in the art would be motivated to eliminate recompilation of a file and to encapsulate data.

Per Claims 23-24 (Amended) & 25:

Fortin teaches a computerized method for instrumenting an original function in an executable file for testing by callers of the original function ("provide a system and method for efficiently instrumenting stripped object routines executing on a computer system" in column 3, lines 19-21), modifying the original function to invoke a user-supplied function in place of the original function, the user-supplied function enabling code to control execution of the original function ("The Call to Target still points to address 202. After 202, however, control is passed to

Entry Routine 210. Entry Routine 210 collects the information desired by the monitor and returns control to the target routine." in column 4, lines 46-51), adding an entry in a function lookup table of the address of the original function; retrieving the entry from the function lookup table using the name of the original function; and invoking the original function using the entry ("insertion of the demultiplexor entries 510. A demultiplexor entry (demux-entry) is provided for each target routine." in column 5, lines 46-51 and lines 64-67; column 6, lines 6-9; and column 6, lines 35-39 and lines 53-58), wherein the modified original function comprises an instruction that causes a jump to the user-supplied function ("Processing begins with a call to the target routine 702. This call will encounter the branch first instruction and immediately branch to the Entry section 704 ... that in turn calls 708 user supplied Entry routine" in column 6, lines 46-51), determining whether the prototype of the original function is correctly specified, and indicating an error when the determining whether the prototype of the original function is correctly specified indicates an incorrectly specified prototype of the original function ("The routines also allow the function of a target routine to be fully replaced such that newly provided code will be executed instead of the base code in the routine being monitored." in column 4, lines 62-65, inherently determining prototype). Fortin does not explicitly teach an embedded function, modifying an embedded function using a wrapper or that the code is fault simulation code.

Official Notice is taken that it was a common practice to include an embedded function at the time the instant invention was made. Kaufer teaches fault simulation code (column 1, lines 59-64). Peek teaches modifying a function using a wrapper ("providing a "wrapper" for the

particular function ... a wrapper will be seen to be an individual piece of code placed around a corresponding individual function call or routine" in column 6, lines 34-37 and 43-50).

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by Fortin to include an embedded function, modifying an embedded function using a wrapper and fault simulation code using the combined teaching of common practice, Kaufer and Peek. The modification would be obvious because one of ordinary skill in the art would be motivated to encapsulate functions.

Per Claim 26:

The rejection of claim 23 is incorporated, and Fortin further teaches that the function lookup table is in the executable file (column 5, lines 46-51).

Response to Arguments

9. Applicant's arguments filed on 11/15/04 have been fully considered but they are not persuasive.

In the remarks, the applicant argues that:

a) The Office Action rejected independent Claim 1 by citing a procedure described in Fortin for monitoring a routine. Claim 1, as amended, teaches "retrieving the access information from the function lookup table using the name of the original function; and invoking the original function using the access information." The method taught by Claim 1 is significantly different from the procedure in Fortin cited by the Office Action.

The procedure in Fortin is directed to the sequential execution of a target routine, an entry routine, and an exit routine when the target routine is called such that the entry and exit routines collect information about the target routine. Neither Fortin, Kaufer, Peek nor any combination thereof disclose anything about "retrieving the access information from the function lookup table using the name of the original function; and invoking the original function using the access information," as recited in Applicant's amended Claim 1. Applicants respectfully submit that the rejection of Claim 1 is overcome and request that the rejection be withdrawn.

Claims 15, 21, 23, 28-31, 36 and 41, as amended, include limitations substantially similar (albeit different in other important ways) to the limitations claimed in the currently amended Claim 1. As discussed above, Claim 1 is allowable. Thus, Claims 15, 21, 23, 28-31, 36 and 41 are allowable for at least the same reasons that Claim 1 is allowable, and notice to that effect is solicited. Furthermore, dependent Claims 2-14, 16-20, 22, 24-26, 32, 33 and 42-47 are allowable for at least the same reasons that the base claims on which they rely are allowable, and notice to that effect is solicited.

Examiner's response:

a) Examiner strongly disagrees with applicant's assertion that Fortin fails to disclose the claimed limitations recited in amended claims 1, 15, 21, 23, 28-31, 36 and 41. Fortin clearly shows each and every limitation in amended claims 1, 15, 21, 23, 28-31, 36 and 41. Fortin teaches retrieving the access information from the function lookup table using the name of the original function; and invoking the original function using the access information ("demultiplexor entries" in column 5, lines 46-67; column 6, lines 6-9; and column 6, lines 35-39

and lines 53-58). In addition, see the rejection above in paragraphs 7 and 8 for rejection to claims 1-26, 28-33, 36 and 41-47.

Conclusion

10. Any inquiry concerning this communication from the examiner should be directed to Qamrun Nahar whose telephone number is (571) 272-3730. The examiner can normally be reached on Mondays through Thursdays from 8:30 AM to 6:00 PM. The examiner can also be reached on alternate Fridays.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki, can be reached on (571) 272-3719. The fax phone number for the organization where this application or processing is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

QN
December 7, 2004

TODD INGBERG
PRIMARY EXAMINER